Using PDDL to Solve Vehicle Routing Problems

Wenjun Cheng, Yuhui Gao

Using PDDL to Solve Vehicle Routing Problems

Wenjun Cheng[1], Yuhui Gao[2]

[1]Academy of Military Science, Beijing 100091, China
berrywen@hotmail.com
[2]Beijing Aerospace Control Centre, Beijing 100094, China
michaeldoer@hotmail.com

Abstract. In this paper, we describe a search method based on the language PDDL for solving vehicle routing problems. The Vehicle Routing Problem (VRP) is a classical problem in Operations Research, and there are many different variants of the VRP. This paper describes a new approach to model standard VRP and some variants based on PDDL language, explains how the method constructs model and solves the problem using several PDDL planners, and analyses the planning results of these planners.

Keywords: Artificial Intelligence Planning, Vehicle Scheduling, PDDL, Vehicle Routing

1 Introduction

The vehicle routing problem is a well-known combinatorial optimization problem in the field of service operations management and logistics [1]. It is described as finding the minimum distance or cost of the combined routes of a number of vehicles that must service a number of customers. The number of feasible solutions for the VRP increases exponentially with the number of customers to be serviced, and almost each of VRP's variants is more complicated than standard VRP. Since there is no known polynomial algorithm that will find the optimal solution, the VRPs is considered NP-hard. For finding solutions of such problems, the use of problem-specific solvers with heuristics is considered a reasonable approach that most of VRP research is based on. However it is realistic to solve VRPs that are structured in the same way using universal planners. This paper describes the methods to model some of the variants of VRP in PDDL. The planners for PDDL are automated, which means that they solve the problem by using heuristics that are pre-developed, meaning that the heuristics do not change depending on the type of problem. So it provides attempt to solve the VRPs through a uniform method.

The remainder of this paper is organized as follows. Section 2 introduces the VRP and its multiple variants; Section 3 describes the features of PDDL language and the main versions of PDDL; Section 4 gives the models of several VRPs based on PDDL; Section 5 provides some efficient PDDL planners to solve the models and compares the planning results. Section 6 concludes with a brief discussion and perspectives.

2 The standard VRP and its Variants

The standard VRP can be described as a weighted graph G=(V, A, d) where $V=\{v_1,v_2,\ldots,v_n\}$ is a set of vertices representing locations with the depot located at v1 and A is the set of arcs. With every arc(v_i, v_j) ($i \neq j$) is associated a non-negative distance d which can be interpreted as a travel cost or as a travel time. The standard VRP consists of designing a set of least-cost vehicle routes in such a way that

- Each customer is visited only once by a single vehicle, and is served according to their demand.
- Each vehicle must start and end its route at the depot v1, and it is not allowed to visit the depot in between.

The standard VRP can be extended in multiple ways which add some constraints to be satisfied. The most common constraints include:

- Capacity restrictions: each vehicle has finite capacity and each location has a finite demand. So the sum of the demands for the vehicle's tour does not exceed its capacity. Capacity-constrained VRPs will be referred to as CVRPs.
- Total time restrictions: the length of any route may not exceed a prescribed bound. Time-constrained or distance-constrained VRPs will be referred to as DVRPs.
- Time windows restrictions: the service at each customer must start within an associate time window and the vehicle must remain at the customer location during service. The VRP with time windows will be referred to as VRPTW.
- Other restrictions including multiple depots, multiple goods, multiple service modes, multiple transport modes, or different optimization goals. So there are a variety of VRP variants such as multiple depot vehicle routing problem (MDVRP), the vehicle routing problem with backhauls (VRPB), the vehicle routing problem with backhauls and time windows (VRPBTW), and VRP with pickup and delivery (VRPPD).

These VRPs were made by industrial requirements and have been studied since 1995. With the simpler variants of the VRP being solved, the VRP research community pays more attention to rich VRP models and larger size problems recently. We prefer to construct some common models to descript most of VRP variants from simple to complex, and choose a uniform way to solve these models. In the field of artificial intelligence (AI), PDDL is an expressive language which offers to represent many planning knowledge while freeing from the plan search algorithms, and PDDL can satisfy the demand of modelling various kinds of VRP.

3 PDDL and its Versions

PDDL, the "Planning Domain Definition Language", is a standard language for expressing planning problems and domains is widely used in the planning community [2]. This enables us to effortlessly select the best suited planner for describing the

technical and epistemological requirements of planning domains and the capabilities of planners in a uniform way. A planning problem is made of an initial state, a final state and a set of operators; a state is a set of predicates and operators can be seen as a couple of states: preconditions which must be true to allow execution and effects which are true after execution. In PDDL, the specification of the predicates and the operator schemas is separated from the specification of the initial state and the goal condition. The former is typically referred to as the planning domain and the latter as the planning problem, so a number of planning problems can be defined for the same planning domain.

There are dozens of different versions of PDDL and its extension. Every version of PDDL specifies new requirements that should be supported and increases the flexibility and functionality of PDDL language. PDDL1.2 states the basics of the language, such as how the domain and problem file should be composed. One of the most important PDDL versions is 2.1, which specifies the possibility to use numerical variables, plan-metrics, and durative actions which make planning with conditions such as fuel consumption, time constraints and capacity constraints possible [3]. In PDDL versions 2.2, derived predicates and timed initial literals which are useful in modelling VRPs are specified [4]. However, for higher PDDL versions, only a few planners provide effective support, and we cannot use the specifications of these PDDL versions.

4 PDDL Models

4.1 PDDL Model for CVRP

In the problem there are three different types: freights, vehicles, and locations.

Planning problems predicates are objects that are described using Boolean representation. The following predicates represent facts related to the types.

- (at freight ?f, location ?l): the freight ?f is currently at the location ?l.
- (at2 vehicle ?v, location ?l): the vehicle ?v is currently at the location ?l.
- (adjacent location ?l1, location ?l2): the location ?l1 is adjacent to ?l2.
- (in freight ?f, vehicle ?v): the freight ?f is currently loaded by the vehicle ?v.
- (using vehicle ?v): the vehicle ?v is not at the depot.
- (destination freight ?f location ?l): the location ?l is the destination of the freight ?f.

Planning problems functions are objects with states that can be represented numerically (also called numerical fluents). The following predicates represent facts related to the types.

- (totalCost): this descripts the variable that keeps track of the total cost of the solution. It is the key function to solve Multi-criterion and multi-objective optimal problems.
- (weight freight ?f): this descripts the weight of the freight ?f.
- (maxLoad): this descripts the maximum load limit of all vehicles.

- (load vehicle ?v): this descripts how much weight the vehicle ?v is carrying.
- (distance location ?l1 location ?l2): this descripts the distance between ?l1 and ?l2. This function is only available for adjacent locations.

Actions are events that change the state of the problem. The available actions in the domain are defined in terms of their preconditions and effects using the available predicated and the functions of the domain. For example, the follow PDDL statement is an action schema as it appears in the CVRP PDDL domain that represents the actions of the vehicle loading the freight.

```
(:action load
  :parameters (?f - freight ?v - vehicle ?l - location)
  :precondition (
     and  (using ?v)
          (at ?f ?l)
          (at2 ?v ?l)
          (>= (maxLoad) (+(load ?v) (weight ?f))))
  :effect (
     and  (in ?f ?v)
          (not (at ?f ?l))
          (increase (cargo ?v) (weight ?f))))
)
```

In this problem there are four actions possible:

- useVehicle(vehicle ?v): set the vehicle ?t as available for use.
  precondition: the vehicle ?v is not in use.
  effect: the vehicle ?v is in use.
- load(freight ?f, vehicle ?v, location ?l): loads the freight ?f into the vehicle ?v at the location ?l.
  precondition: the vehicle ?v is in use; it and the freight ?f are both at the tion ?l; maxLoad >= $load_v$ + $weight_f$.
  effect: the freight ?f is in the vehicle ?v and is not at the location ?l; $load_v$=$load_v$+$weight_f$.
- unload(freight ?f, vehicle ?v, location ?l): unloads the freight ?f from the vehicle ?v at the location ?l.
  precondition: the vehicle ?v is in use and at the location ?l where is the destination of the freight ?f; ?f is in the vehicle ?v.
  effect: the freight ?f is at the location ?l and is not in the vehicle ?v; $load_v$=$load_v$-$weight_f$.
- move(vehicle ?v, location $?l_1$, location $?l_2$): moves the vehicle ?v from $?l_1$ to $?l_2$.
  precondition: the vehicle ?v is in use and at the location $?l_1$; $?l_1$ is adjacent to $?l_2$.
  effect: the vehicle ?v is at $?l_2$ and not at $?l_1$.

Inside the problem file, all types of planning objects are defined and initialized, and the planning goal is set. The domain that is described above can have many VRP problem files, which might differ by the number of objects, initial values and goal.

For CVRP PDDL domain, all freights should be at their respective goal location and all vehicles should have returned to their respective origin. The function "totalCost" should be minimized which is the metric of the problem and always equal to the objective function.

Some PDDL features need to be supported in constructing PDDL models for CVRP: typing, fluents, plan metrics. These features are specified before PDDL version 2.1, so the models can be solved by any planner which supports PDDL2.1.

4.2 PDDL Models for DVRP and VRPTW

In the PDDL model for DVRP, it is necessary to convert the transport of vehicles to durative actions. So the PDDL feature "durative actions" is added to the feature list which is must be supported in the PDDL model.

Some correspond changes must be applied to the former PDDL model for CVRP. For example, the follow PDDL statement is a durative action schema that represents the actions of the vehicle loading the freight.

```
(:durative-action load
  :parameters (?f - freight ?v - vehicle ?l - location)
  :duration(= ?duration (loadTime ?f))
  :condition (
    and (over all(using ?v))
        (at start(at ?f ?l))
        (at start(at2 ?v ?l))
        (at start(>= (maxLoad) (+ (load ?v)(weight ?f)))))
  :effect (
    and (at end(in ?f ?v))
        (at start(not (at ?f ?l)))
        (at end(increase (load ?v) (weight ?f)))))
)
(loadTime freight ?f): this function descripts how long
      the freight ?f is loaded or unloaded by vehicles.
```

In VRPTW, there are individual time windows for each customer, and possibly also on the depot. The PDDL feature "timed initial literals" has the possibility to make predicates available according to predetermined times and consequently to make actions feasible at special time. This feature is specified in PDDL 2.2 and only a few planners support it. Only one predicate wants to be added to the domain file of DVRP model to support time windows.

```
(available ?f - freight) : descripts the freight ?f
       available(depend on time windows).
```

The PDDL planning problem generated for above modes is the following (most of the predicates should be self-explanatory):

```
(define (problem VRPP)
```

```
  (:domain VRP)
  (:objects f1 - freight f2 - freight
            v1- vehicle v2- vehicle
            l1 - location l2 - location l3 - location)
  (:init
    (adjacent l1 l2) (adjacent l2 l1)
    (adjacent l1 l3) (adjacent l3 l1)
    (= (distance l1 l2) 22) (= (distance l2 l1) 22)
    (= (distance l1 l3) 25) (= (distance l3 l1) 25)
    (= (weight f1) 8) (= (weight f2) 9)
    (= (load v1) 0) (= (load v2)0)
    (at 0 (available f1)) (at 100 (not (available f1)))
    (at 0 (available f2)) (at 100 (not (available f2)))
    (= (maxLoad) 10) (= (totalCost) 0)
    (at f1 l2) (at f2 l3) (at2 v1 l1) (at2 v2 l1)
    (destination f1 l1) (destination f2 l1)
  )
  (:goal
    (and (at f1 l1) (at f2 l1) (at2 v1 l1) (at2 v2 l1))
  )
  (:metric minimize (totalCost))
)
```

4.3 PDDL Models for Other VRPs

For other VRPs, such as MDVRP, VRPB, VRPBTW and VRPPD, the different from above VRPs is focused on multiple vehicle types, multiple service types, route restrictions, and freight constraints. It doesn't need more PDDL features to describe the PDDL model for these VRPs. All we need to do is providing more predicates, functions and actions.

5 Performance of PDDL Planners

The PDDL planners are automated and the heuristics search algorithms embedded in them does not change depending on the type of problem. There are many PDDL planners available on the internet, with certain characteristics that may make them more or less suitable for different kinds of planning problems.

The planners we tried were SAPA [5], LPG-TD [6-8], SGPlan6 [9], TFD [10], POPF2 [11] and DaeYa [12]. All of these planners can solve the DVRP problems because they have the ability to satisfy the "durative-action" feature of PDDL 2.1 and process the problems which contain durative actions. But only SAPA, LPG-TD, POPF2 and SGPlan support the "timed-initial-literals" feature of PDDL 2.2 and can solve the VRPTW problems.

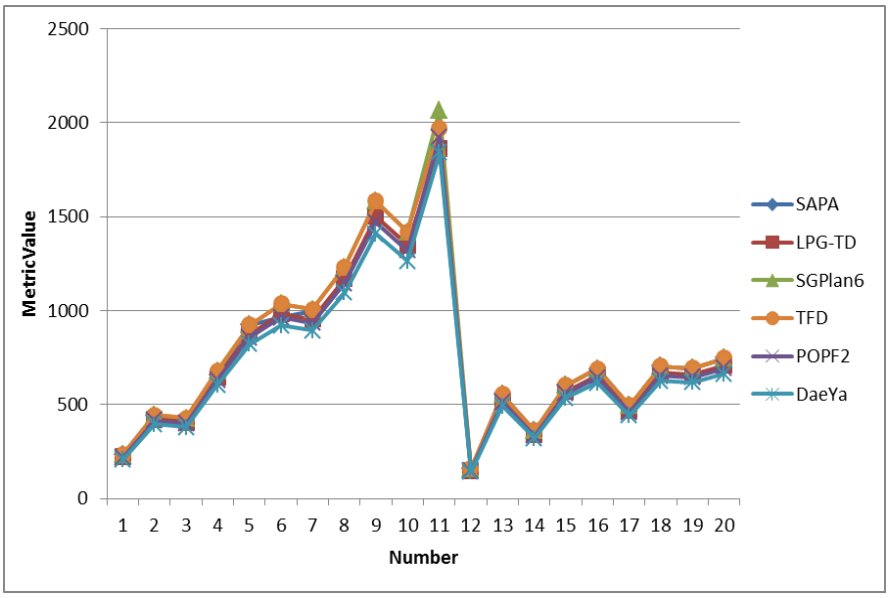

Fig. 1. DVRP Benchmark

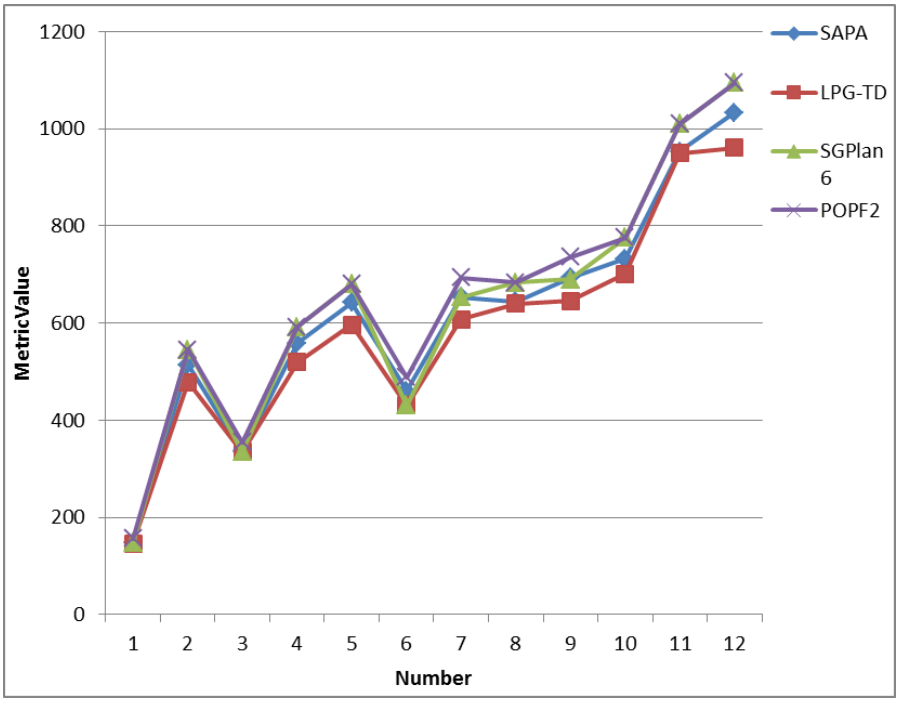

Fig. 2. VRTTW Benchmark

Figure 1 shows the performance of planners on DVRP benchmark instances. 20 DVRP instances used range from between 10 to 30 locations, 10 to 50 freights, and from 10 to 20 vehicles which are auto-generated using the code in Matlab.

Figure 2 shows the performance of planners on VRPTW benchmark instances. 12 VRPTW instances used range from between 10 to 20 locations, 10 to 30 freights, and from 10 to 20 vehicles which are auto-generated using the code in Matlab.

DaeYa planner performs the best of the planners in the DVRP case and LPG-td planner performs the best of the planners in the VRPTW case. We only consider the small-scale VRP problems because the performances of most PDDL planners are not the same level as problem-specific solvers.

6 Conclusions

This paper describes a new approach to model standard VRP and some variants based on PDDL language, explains how the method constructs model and solves the problem using several PDDL planners, and analyses the planning results of these planners. Comparing to problem-specific solvers, this method provides common models to describe VRP problems and uniform solving mode by using different planners. Although the performances of multiple PDDL planners are not the same level as problem-specific solvers, it is realistic to expect planners to solve problems that are structured in the same way as real-world problems.

References

1. Laporte, G.: Fifty Years of Vehicle Routing. In: Transportation Science, vol. 43, pp. 408-416 (2009).
2. McDermott, D., Ghallab, M.: PDDL-The Planning Domain Definition Language. Technical Report, CVC TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control, (1998).
3. Fox, M., Long, D.: PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains. In: Journal of Artificial Intelligence Research (JAIR), vol. 20, pp. 61–124 (2003).
4. Edelkamp, S., Hoffmann, J.: PDDL2.2: The Language for the Classical Part of the 4th International planning Competition. Technical Report, the 4th International planning Competition (2004).
5. Do, M.B., Kambhampati, S.: SAPA: A multi-objective metric temporal planner. In: Journal of Artificial Intelligence Research (JAIR), vol. 20, pp. 155-194 (2003).
6. Gerevini, A., Serina, I.: LPG: A planner based on a local search for planning graphs with action costs. In: 6th International Conference on Artificial Intelligence Planning and Scheduling (AIPS'02), pp. 13-22 (2002).
7. Gerevini, A., Saetti, A.., Serina, I: Integrating Planning and Temporal Reasoning for Domains with Durations and Time Windows. In: 19th International Joint Conference On Artificial Intelligence (2005).
8. Gerevini, A., Saetti, A., Serina, I.: An approach to temporal planning and scheduling in domains with predictable exogenous events. In: Journal of Artificial Intelligence Research (JAIR), vol. 25, pp. 187-231 (2006).

9. Chen, Y., Benjamin, W.Wah, Chih-Wei Hsu: Temporal planning using subgoal partitioning and resolution in SGPlan. In: Journal of Artificial Intelligence Research, vol. 26, pp. 323-369 (2006).
10. Eyerich, P., Mattmüller, R., Röger, G.: Using the context enhanced additive heuristic for temporal and numeric planning. In: 19th International Conference on Automated Planning and Scheduling (ICAPS), Thessaloniki, Greece (2009).
11. Coles, A. J., Coles, A., Fox, M., Long, D.: Forward chaining partial-order planning. In: 20th International Conference on Automated Planning and Scheduling (ICAPS), pp. 42-49, Toronto, Ontario, Canada (2010).
12. Dréo, J., Savéant, P., Schoenauer, M., Vidal, V.: Divide-and-Evolve: The marriage of Descartes and Darwin. In: 2011 International Planning Competition (2011).